

注意：所有前端传递的数据如添加、登录、注册等至少进行简单的非空验证；

# 邻家书房接口文档

## 1. 登录接口

接口说明：用户登录；接收手机号，密码；返回 token

请求方法：如无特别说明，统一为 get 请求

请求 URL：<http://192.168.1.180:8080/user/login>

请求参数说明：手机号：Long phone, 密码：String password

请求参数示例：

<http://192.168.1.180:8080/user/login?&phone=123&password=123>

返回值说明：返回 token 值，token 值在 cookie 中对应的 key 是 TT\_TOKEN

返回值示例：

```
{"status":200,"msg":"OK","data":"a87ca3fb-3c41-4188-a1a4-daf0fd030568"}
```

## 2. “我的” 页面接口

接口说明：通过 token 查询用户信息

请求方法：如无特别说明，统一为 get 请求

请求 URL：<http://192.168.1.180:8080/user/token>

请求参数说明：token 值

请求参数示例：

<http://192.168.1.180:8080/user/token/a87ca3fb-3c41-4188-a1a4-daf0fd030568>

返回值说明：返回包装用户信息的实体类 GuiResult

返回值示例：

```
{
  "status":200,
  "msg":"OK",
  "data":{
    "id":3,
    "phone":123,
    "password":null,
    "nickname":"千万人",
    "gender":"男",
    "age":25,
    "profession":"萨芬",
    "motto":"如果是",
    "address":"水电费",
  }
}
```

```

        "image":null,
        "name":"去热饭",
        "images":null,
        "created":1530003597000,
        "updated":1530003599000
    }
}

```

### 3.我的心得页面接口

接口说明：通过用户 ID 查询用户心得并展示到页面

请求方法：如无特别说明，统一为 get 请求

请求 URL：<http://192.168.1.180:8080/my/all>

请求参数说明：用户 ID: Long userId

请求参数示例：

<http://192.168.1.180:8080/my/all?&userId=2>

返回值说明：返回包装用户心得的实体类 GuiResult

返回值示例：

```

{
    "status":200,
    "msg":"OK",
    "data":[
        {
            "id":3,
            "userId":null,
            "title":"杜拉拉升级",
            "content":"一刀 999",
            "created":"2018-07-02 17:09:18.0",
            "updated":"2018-07-02 17:09:20.0"
        },
        {
            "id":4,
            "userId":null,
            "title":"热议和",
            "content":"微文深诋",
            "created":"2018-07-03 14:29:04.0",
            "updated":"2018-07-03 14:29:06.0"
        }
    ]
}

```

## 4.进入书房页面接口

接口说明：进入书房页面接口，接收用户 ID，获取跟该用户相关的用户基本信息、书房信息、书籍信息，用 GUIresult 包裹返回。

请求方法：如无特别说明，统一为 get 请求

请求 URL：<http://192.168.1.180:8080/my/study>

请求参数说明：用户 ID: Long userId

请求参数示例：

<http://192.168.1.180:8080/my/study?&userId=2>

返回值说明：如无特别说明，统一使用 GuiResult 实体类包裹数据进行返回，具体格式如下

返回值示例：

```
{
  "status":200,
  "msg":"OK",
  "data":{
    "用户年龄":21,
    "用户地址":"大概",
    "用户书房照片":null,
    "用户昵称":"小王",
    "用户头像":null,
    "用户座右铭":"驱蚊器群",
    "用户书房名":"小王搬砖卡",
    "用户性别":"男",
    "出借书单":[
      {
        "id":37,
        "bookname":null,
        "author":null,
        "imges":null,
        "isbn":null,
        "pubname":null,
        "pubtime":null,
        "editionorder":null,
        "pages":null,
        "language":null,
        "ownerId":null,
        "created":null,
        "updated":null
      }
    ],
    "出售书单":[
      {
        "id":40,
```

```

        "bookname":null,
        "author":null,
        "imges":null,
        "isbn":null,
        "pubname":null,
        "pubtime":null,
        "editionorder":null,
        "pages":null,
        "language":null,
        "ownerId":null,
        "created":null,
        "updated":null
    }
}
]
}
}

```

## 5.我的借出页面接口

接口说明：我的借出页面接口，接收用户 ID，查询用户出借的书中有哪些已被借走及借出的相关信息，返回值使用 GuiResult 包裹。

请求方法：如无特别说明，统一为 get 请求

请求 URL: <http://192.168.1.180:8080/my/LoanOut>

请求参数说明：用户 ID: Long userId

请求参数示例：

<http://192.168.1.180:8080/my/LoanOut?&userId=2>

返回值说明：具体参数的含义，等你来我就告诉你，啊哈哈！返回值示例：

```

{
    "status":200,
    "msg":"OK",
    "data":[
        {
            "id":40,
            "bookname":"开了房",
            "author":"耳环",
            "imges":null,
            "loanId":1,
            "nickname":"小微",
            "name":"小微书房",
            "status":2,
            "loanTime":10,
            "updated":1530414701000,
            "lastStatus":"剩余还书时间：5 天 20 时 14 分"
        },
    ],
}

```

```

{
  "id":37,
  "bookname":"神雕侠侣",
  "author":"金庸",
  "imges":null,
  "loanId":4,
  "nickname":"太让人",
  "name":"好地方",
  "status":3,
  "loanTime":2,
  "updated":1530414700000,
  "lastStatus":"还书状态: 未还书 （超时 2 天 3 时 45 分）"
}
]
}

```

## 6.我的借入页面接口

接口说明：我的借入页面接口：接收用户 ID，根据该 ID 查看用户借了哪些书，借书信息以及该书拥有者的相关信息

请求方法：如无特别说明，统一为 get 请求

请求 URL：<http://192.168.1.180:8080/my/borrow>

请求参数说明：用户 ID: Long userId

请求参数示例：

<http://192.168.1.180:8080/my/borrow?&userId=3>

返回值说明：如无特别说明，统一使用 GuiResult 实体类包裹数据进行返回，具体格式如下

返回值示例：

```

{
  "status":200,
  "msg":"OK",
  "data":[
    {
      "id":39,
      "bookname":"放假放假",
      "author":"后端",
      "imges":null,
      "ownerId":5,
      "nickname":"将获得丰厚的",
      "name":"哥啊",
      "status":4,
      "loanTime":3,
      "updated":1530676554000,
      "lastStatus":"还书状态: 已还书"
    },

```

```

{
  "id":38,
  "bookname":"去玩儿",
  "author":"请问",
  "imges":null,
  "ownerId":1,
  "nickname":"小微",
  "name":"小微书房",
  "status":3,
  "loanTime":20,
  "updated":1529985359000,
  "lastStatus":"剩余还书时间：10 天 20 时 53 分"
}
]
}

```

## 7.微博授权登录接口

接口说明：用户点击微博登录图标，前端请求新浪微博授权登录 URL，授权成功以后前端拿到 code 传递给后台

请求方法：如无特别说明，统一为 get 请求

请求 URL：<http://192.168.1.180:8080/weibo/auth>

请求参数说明：code

请求参数示例：

<http://192.168.1.180:8080/weibo/auth?code=65fca80b8991e6a04f8e316dc8aa6a43>

返回值说明：三种情况（1、用户第一次使用微博登录我方 APP，让他去绑定账号；2、用户不是第一次用微博登录 APP 并且已经绑定账号，返回 token，跳转首页；3、后台处理发生异常）

返回值示例：

情况 1：

```

{"msg":"请进行账号绑定",
"openID":"6592302873",
"accessToken":"2.00hHdIMH6qKICB987fc6489eVkJuDD",
"status":300}

```

情况 2：

```

{"msg":"微博授权登录成功",
"status":200,
"token":"538f8908-dad8-45e0-a3bd-eb9034ce4cbc"}

```

情况 3：

```

{"status":502,"msg":"微博登录发生异常"}

```

## 8.微博账号绑定接口

接口说明：用户输入手机号和密码进行账号绑定

请求方法：post 请求

请求 URL: `http://localhost:8080/weibo/binding`

请求参数说明: 电话号码 Long phone, 密码 String password,上面接口情况 1 返回得到的 String accessToken 和 String openID

请求参数示例:

`http://localhost:8080/weibo/binding`

返回值说明: 返回 json 格式的 map 集合

返回值示例:

情况 1: 用户输入的手机号不存在或者错误, 让他去注册或者重新输入

`{"data":null,"msg":"该用户未注册或手机号错误","status":301}`

情况 2: 用户输入的密码错误

`{"data":null,"msg":"手机号或密码错误, 请重新输入! ","status":302}`

情况 3: 手机号和密码都没问题, 返回 token 值, 跳转首页

`{"data":"2caecf4b-5375-4f75-94c8-1fae65f577fe","msg":"OK","status":200}`

情况 4: 后台操作异常

`{"data":null,"msg":"绑定账号出现异常! ","status":303}`

## 9、QQ 授权登录

请求 URL: `http://192.168.1.180:8080/qq/login`

其他参考微博登录

## 10、QQ 账号绑定

请求 URL: `http://localhost:8080/qq/binding`

其他参考微博账号绑定

## 11、微信授权登录

请求 URL: `http://192.168.1.180:8080/weixin/login`

其他参考微博登录

## 12、微信账号绑定

请求 URL: `http://localhost:8080/weixin/auth`

其他参考微博账号绑定

## 13、图片上传接口

接口说明: 上传图片, 返回图片 URL

请求方法: 如无特别说明, 统一为 get 请求

请求 URL: `http://localhost:8080/upload`

请求参数说明：上传图片

请求参数示例：

无

返回值说明：string 类型的 URL

返回值示例：

无

## 14、通过 token 查询用户信息接口

接口说明：接收 token 调用 Service 返回用户信息，使用 GuiResult 包装。

请求方法：如无特别说明，统一为 get 请求

请求 URL：http://localhost:8080/user/token/{token}

请求参数说明：用户登录成功以后返回的 token 作为请求参数

请求参数示例：

http://localhost:8080/user/token/a468cd06-9eff-470d-8283-234503e37ec7

返回值说明：如果用户 token 不过期则返回用户信息

返回值示例：

```
{"data":{"address":"水电费","age":25,"created":"2018-06-26 16:59:57","gender":"男","id":3,"image":null,"images":null,"motto":"如果是","name":"去热饭","nickname":"千万人","password":null,"phone":123,"profession":"萨芬","updated":"2018-06-26 16:59:59"},"msg":"OK","status":200}
```

## 15、短信验证码接口

接口说明：发送手机验证码

请求方法：post

请求 URL：http://localhost:8080/user/send/message

请求参数说明：mobile 表示手机号；sig 是签名，签名=MD5（手机号+秘钥）；秘钥是 cc6ecf659073e51aa8291d686458a4ce，例如 18312522890cc6ecf659073e51aa8291d686458a4ce 经过 MD5 加密以后就是 8e8888782d48025e552380ada7905bec

请求参数示例：

sig=8e8888782d48025e552380ada7905bec

mobile=18312522890

返回值说明：正常返回 status：200；

返回错误码示例：111：“手机号格式不正确”；123，“错误请求：签名验证不通过”；“该手机号 120 秒以内只能接收一次验证码！”等等。

返回值示例：

```
{"data":null,"msg":"OK","status":200}
```



## 16、注册接口

接口说明：接收手机号，密码，验证码，验证通过插入数据库，注册完毕跳转登录页面重新登录

请求方法：如无特别说明，统一为 get 请求

请求 URL: `http://localhost:8080/user/register`

请求参数说明：mobile 手机号码；password 密码；code 短信验证码

请求参数示例：

`http://localhost:8080/user/register?mobile=18312522890&password=123&code=22925`

返回值说明：验证码无误返回状态 200

返回值示例：

`{"data":null,"msg":"OK","status":200}`

## 17、忘记密码接口

接口说明：接收手机号和验证码，验证通过进入“下一步”

请求方法：如无特别说明，统一为 get 请求

请求 URL: `http://localhost:8080/user/miss/password`

请求参数说明：mobile 手机号码；code 短信验证码

请求参数示例：

`http://localhost:8080/user/register?mobile=18312522890&code=22925`

返回值说明：验证码无误返回状态 200

返回值示例：

`{"data":null,"msg":"OK","status":200}`

## 18、修改密码接口

接口说明：接收手机号和密码，后台修改密码

请求方法：如无特别说明，统一为 get 请求

请求 URL: `http://localhost:8080/user/update/password`

请求参数说明：mobile 手机号码；password 要修改的密码

请求参数示例：

`http://localhost:8080/user/update/password?mobile=456789&password=456789`

返回值说明：验证码无误返回状态 200

返回值示例：

`{"data":null,"msg":"OK","status":200}`

## 19、修改个人信息接口

接口说明：用 user 对象接收 userID，性别，年龄等，插入数据库

请求方法：如无特别说明，统一为 get 请求

请求 URL: <http://localhost:8080/user/modify/personal>

请求参数说明: 表单内容 id 代表用户 ID, nickname 昵称、gender 性别、age 年龄, profession 用户职业、motto 座右铭、address 地址、image 用户头像、name 书房名、images 书房照片

请求参数示例:

无

返回值说明: 验证码无误返回状态 200

返回值示例:

```
{"data":null,"msg":"OK","status":200}
```

## 20、添加图书

接口说明: 接收图书相关信息, 图书所有者 id, 图书类型(1 代表出借, 2 代表出售), 图书价格(出借每天多少钱, 出售每本多少钱)

请求方法: 如无特别说明, 统一为 get 请求

请求 URL: <http://localhost:8080/book/add/book>

请求参数说明: 表单内容 bookname 代表书名, author 作者、imges 图书图片、isbn 代表 isbn 号、pubname 出版社、pubtime 出版时间、editionorder 版次、pages 总页数、language 正文语言、userId 用户 id、type (图书类型)、price 图书价格

请求参数示例:

无

返回值说明: 验证码无误返回状态 200

返回值示例:

```
{"data":null,"msg":"OK","status":200}
```

## 21、图书详情接口

接口说明: 当用户点击一个图书的图片后会跳转详情页, 接收图书 id, 返回与图书相关的信息

请求方法: 如无特别说明, 统一为 get 请求

请求 URL: <http://localhost:8080/book/details>

请求参数说明: bookId 图书 id

请求参数示例:

<http://localhost:8080/book/details?bookId=37>

返回值说明: 验证码无误返回状态 200

返回值示例:

```
{"data":{"book":{"author":"金庸","bookname":"神雕侠侣", "created":"2018-07-04 11:52:11","editionorder":"12355","id":37,"imges":null,"isbn":"123465", "language":"汉语","ownerId":2,"pages":1000,"pubname":"江南出版社", "pubtime":"2000 年","updated":"2018-07-04 11:52:14"},"user":{"address":"大概","age":21,"created":"2018-06-25 16:59:17","gender":"男","id":2,"image":null,"images":null,"motto":"驱蚊器群","name":"小王搬砖卡","nickname":"小王"}}
```

```

", "password": "456789", "phone": 456789, "profession": "搬砖的",
", "updated": "2018-06-25 16:59:21"}}, "msg": "OK", "status": 200}

```

## 22、提交订单接口

接口说明：接收前端表单提交的内容，返回订单号和应付总金额

请求方法: post

请求 URL: <http://localhost:8080/book/create/order>

请求参数说明：**payment** 实付金额、**userId** 买家 ID、**sellerId** 卖家 ID、**receiverAreaName** 收货地址、**receiverMobile** 收货手机号、**receiver** 收货人姓名、**itemId**、商品 ID、**bookName** 图书名、**price** 图书单价、**image** 图书图片、**num** 购买数量、**totalFee** 商品总金额、**sellerId** 卖家 ID

请求参数示例:

无

返回值说明：返回订单号和应付总金额

返回值示例:

无

## 23、支付接口

待测试

## 24、首页展示数据查询

接口说明：登录成功后，跳转到首页时加载

请求方法: post/get

请求 URL: <http://localhost:8080/comment/findAllList>

参数: `userId` 用户 ID

Coordinate 经度和纬度用", "分割格式示例: 104.394729, 31.125698

请求实例：  
<http://localhost:8080/comment/findAllList?userId=1&Coordinate=104.394729,31.125698>

返回值说明: {

"recommend": [{title: 标题, url: 书 id, pic: 图片路径}。。。]}]为你推荐,书的集合

**"ranking":** [{title: 标题, url: 书 id, pic: 图片路径}{。。。。}]排行  
榜,书的集合

**"newbook":** [{title: 标题, url: 书 id, pic: 图片路径}{。。。。}]新书上市,书的集合

"hobby": [{title: 标题, url: 书 id, pic: 图片路径}{。。。}]]共同爱好, 书的集合

"celebrity": [{title: 标题, url: 书 id, pic: 图片路径}{。。。}]]名人书房的集合

"nearby": [{title: 标题, url: 书 id, pic: 图片路径, "distance": 距离单位米}{。。。}]]附近书房的集合

"friend": [{title: 标题, url: 书 id, pic: 图片路径}{。。。}]]好友书房的集合

}

返回值示例:

```
{
  "celebrity": [{
    "categoryId": 8,
    "id": 16,
    "pic": null,
    "sortOrder": null,
    "status": "1",
    "title": "ygy",
    "url": null
  }],
  "friend": [{
    "categoryId": 7,
    "id": 18,
    "pic": "77",
    "sortOrder": null,
    "status": "1",
    "title": "777",
    "url": "77"
  }, {
    "categoryId": 7,
    "id": 19,
    "pic": "77777",
    "sortOrder": null,
    "status": "1",
    "title": "77777",
    "url": "77777"
  }],
  "recommend": [{
    "categoryId": 2,
    "id": 12,
    "pic": "22",
    "sortOrder": null,
    "status": "1",
```

```
    "title": "22",
    "url": "11"
  }, {
    "categoryId": 2,
    "id": 13,
    "pic": "22",
    "sortOrder": null,
    "status": "1",
    "title": "22",
    "url": "11"
  }],
  "ranking": [{
    "categoryId": 3,
    "id": 14,
    "pic": "22",
    "sortOrder": null,
    "status": "1",
    "title": "33",
    "url": "11"
  }],
  "newbook": [{
    "categoryId": 4,
    "id": 15,
    "pic": null,
    "sortOrder": null,
    "status": "1",
    "title": "44",
    "url": null
  }],
  "nearby": [{
    "distance": "2466",
    "pic": "我的书房照片 URL",
    "title": "我的书房名",
    "url": "1"
  }, {
    "distance": "4384",
    "pic": "sssss",
    "title": "aaa",
    "url": "4"
  }],
  "hobby": [{
    "categoryId": 9,
    "id": 17,
    "pic": null,
```

```

        "sortOrder": null,
        "status": "1",
        "title": "ygytu",
        "url": null
    }]
}

```

## 25、好友动态查询

接口说明：点击好友动态触发，返回好友动态集合-----待测试

请求方法：post/get

请求 URL: <http://localhost:8080/dynamic/friendDynamic>

参数： page, 第几页  
size, 每页显示几条  
userId, 用户 id

返回值说明： [

```

{
  body: 动态的内容
  Time: 发表动态的时间距离现在的时间
  User: 动态的所属用户{id: 用户 id,nickname: 昵称, ...}
  collectionSum: 动态收藏的总数
  userCollection: true 查询用户是否收藏了动态
  commentSum: 动态评论的总数--不包括回复信息
  loveSum: 动态点赞的总数
  userLove: true 查询用户是否点赞了动态
}
{...}

```

]

## 26、单人动态查询

接口说明：点击好友动态触发，返回好友动态集合-----待测试

请求方法：post/get

请求 URL: <http://localhost:8080/dynamic/oneUserDynamic>

参数： page, 第几页  
size, 每页显示几条  
userId, 用户 id  
toId 被查询用户的 id

返回值说明： [

```

{
  body: 动态的内容

```

```

    Time: 发表动态的时间距离现在的时间
    User: 动态的所属用户{id: 用户 id,nickname: 昵称, ...}
    collectionSum: 动态收藏的总数
    userCollection: true 查询用户是否收藏了动态
    commentSum:动态评论的总数--不包括回复信息
    loveSum:动态点赞的总数
    userLove:true 查询用户是否点赞了动态
  }
  {...}
]

```

## 27、圈子动态查询

接口说明: 点击圈子动态触发, 返回圈子动态集合-----待测试

请求方法: post/get

请求 URL: <http://localhost:8080/dynamic/circleDynamic>

参数:

circleid, 圈子 id  
 page, 第几页  
 size, 每页数量  
userId, 用户 id, 根据用户 id 查询这个用户是否点赞...

返回值说明: [

```

{
  body: 动态的内容
  Time: 发表动态的时间距离现在的时间
  User: 动态的所属用户{id: 用户 id,nickname: 昵称, ...}
  collectionSum: 动态收藏的总数
  userCollection: true 查询用户是否收藏了动态
  commentSum:动态评论的总数--不包括回复信息
  loveSum:动态点赞的总数
  userLove:true 查询用户是否点赞了动态
}
{...}
]

```

## 28、根据 id 删除动态

接口说明: 点击动态删除触发, 返回操作是否成功, 提示信息

请求方法: post/get

请求 URL: <http://localhost:8080/dynamic/deleteDynamic>

参数:

id: 动态的 id

返回值说明: {

success:boolean 类型//是否成功

Message:字符串//提示信息

}

## 29、发布动态

接口说明: 点击发布动态触发, 返回操作是否成功, 提示信息

请求方法: post

请求 URL: <http://localhost:8080/dynamic/addDynamic>

参数:

Dynamic dynamic 对象{

body; 发布的内容

circleId; 动态所属分类

userId; 发布者 id

}

返回值说明: {

success:boolean 类型//是否成功

Message:字符串//提示信息

}

## 30、动态点赞

接口说明: 点击点赞, 如果用户已经点赞过了则触发取消点赞, 如果没有点赞则点赞, 返回操作是否成功, 提示信息

请求方法: post、get

请求 URL: <http://localhost:8080/dynamic/addLove>

参数:

Love love 对象{

dynamicId;//动态 id

userId;//点赞人的 id

}

返回值说明: {

success:boolean 类型//是否成功

Message:字符串//提示信息

}

如 需 要 直 接 触 发 取 消 点 赞 请 求 URL :

<http://localhost:8080/dynamic/deleteLove>

参数和返回值同上



## 31、动态收藏

接口说明：点击收藏，如果用户已经收藏过了则触发取消收藏，如果没有点赞则收藏，  
返回操作是否成功，提示信息

请求方法：post、get

请求 URL: <http://localhost:8080/dynamic/addCollection>

参数：

```
Collection collection 对象{  
    dynamicId; //动态 id  
    userId; //点赞人的 id  
}
```

返回值说明：{

```
    success:boolean 类型//是否成功  
    Message:字符串//提示信息
```

}

如 需 要 直 接 触 发 取 消 收 藏 请 求 URL :

<http://localhost:8080/dynamic/deleteCollection>

参数和返回值同上

## 32、动态评论

接口说明：对动态发表评论，或者对评论进行回复

请求方法：post、get

请求 URL: <http://localhost:8080/dynamic/addComment>

参数：

```
Comment comment 对象{  
    dynamicId; 动态 id  
    userId; 发布评论用户 id  
    user2Id; 被评论用户 id  
    body; 评论的内容  
}
```

返回值说明：{

```
    success:boolean 类型//是否成功  
    Message:字符串//提示信息
```

}

## 33、动态获取评论

接口说明：查询动态的所有评论，并按照发布时间排序，

请求方法：post、get

请求 URL: <http://localhost:8080/dynamic/selectComment>

参数:

dynamicId 动态 id

返回值说明: [{

dynamicId; 动态 id

userId; 发布评论用户 id

user2Id; 被评论用户 id

body; 评论的内容

createTime; Date 类型 发布时间

}]

## 34、关注和取消关注

接口说明: 关注或者取消关注, 如果是关注状态触发将自动取消关注

请求方法: post、get

请求 URL: <http://localhost:8080/dynamic/addAttention>

参数:

Attention attention 对象{

attentionId; // 被关注用户 id

userId; // 点关注用户 id

}

返回值说明: {

success: boolean 类型 // 是否成功

Message: 字符串 // 提示信息

}

如 需 要 直 接 触 发 取 消 收 藏 请 求 URL :

<http://localhost:8080/dynamic/deleteAttention>

参数和返回值同上

## 35、查询关注

接口说明: 查询某个人的关注列表

请求方法: post、get

请求 URL: <http://localhost:8080/dynamic/selectAttention>

参数:

Attention attention 对象{

userId; // 被查询用户 id

}

返回值说明: [{

nickName: 用户昵称

Images: 用户书房照片 URL

Name: 用户书房名字  
Id: 用户 id  
}]

## 36、查询钱包

接口说明: 查询自己的钱包,

请求方法: post、get

请求 URL: <http://localhost:8080/wallet/selectAttention>

参数:

userId; // 被查询用户 id

返回值说明: {

id; 用户 id

money; 可用余额

fakeMoney; 暂不可用余额

aliPay; 支付宝账号

payName; 支付宝账号对应的姓名

}

## 37、绑定支付宝, 或者修改已经绑定的支付宝账号

接口说明: 绑定支付宝账号, 或修改已经绑定的支付宝账号

请求方法: post、get

请求 URL: <http://localhost:8080/wallet/updateAliPay>

参数:

walletId; // 钱包 id 同用户 id

aliPay 支付宝账号

payName 真实姓名

返回值说明: {

success: boolean 类型 // 是否成功

Message: 字符串 // 提示信息

}

## 38、提现申请

接口说明: 点击提现申请

请求方法: post、get

请求 URL: <http://localhost:8080/wallet/withdraw>

参数:

walletId; // 钱包 id 同用户 id

Money 申请提现的金额不得低于一元

返回值说明: {  
    **success: boolean** 类型//是否成功  
    **Message: 字符串**//提示信息  
}

## 39、确认收货

接口说明: 点击确认收货

请求方法: post、get

请求 URL: <http://localhost:8080/wallet/withdraw>

参数:

**userId**; //用户 id  
**orderId**; 订单 id

返回值说明: {  
    **success: boolean** 类型//是否成功 **tru** 表示操作成功  
    **Message: 字符串**//提示信息  
}

## 40、获取卖家订单

接口说明: 点击我是卖家, 查询自己作为卖家的订单

请求方法: post、get

请求 URL: <http://localhost:8080/order/findSellerOrdet>

参数:

**sellerId**; //用户 id

返回值说明: [{  
    **orderId**: 订单 id  
    **payment**: 订单金额  
    **Status**: 订单状态 状态: 1、未付款, 2、已付款未发货, 3、已发货, 4、  
        交易成功, 5、交易关闭  
    **consignTime**: 发货时间  
    **shippingName**: 订单物流名  
    **shippingCode**: 订单物流单号  
    **receiverAreaName**: 订单收货地址  
    **receiverMobile**: 订单手机  
    **Receiver**: 订单收货人  
    **image**: 订单图片  
    **bookName**: 订单 shu 名字  
    **price**: 订单单价

```
        num: 订单数量
    }
    {...}]
```

## 41、获取买家订单

接口说明：点击我是卖家，查询自己作为卖家的订单

请求方法：post、get

请求 URL: <http://localhost:8080/order/findBuyerOrder>

参数：

`buyerid`;//用户 id

返回值说明： [{

`orderId`: 订单 id

`payment`: 订单金额

`Status`: 订单状态 状态: 1、未付款, 2、已付款未发货, 3、已发货, 4、交易成功, 5、交易关闭

`consignTime`: 发货时间

`shippingName`: 订单物流名

`shippingCode`: 订单物流单号

`receiverAreaName`: 订单收货地址

`receiverMobile`: 订单手机

`Receiver`: 订单收货人

`image`: 订单图片

`bookName`: 订单 shu 名字

`price`: 订单单价

`num`: 订单数量

```
    }
    {...}]
```

## 42、搜索

接口说明：输入搜索条件，点击搜索，可根据书名模糊查询，其他条件需要完全匹配

请求方法：post、get

请求 URL: <http://localhost:8080/search/conditionSearch>

参数：

`search`;//查询条件，可以输入书名、作者、书房号

返回值说明： {

`searchBook`: [{

`bookname`: 书名

`author`: 作者

`imges`: 图片

`pubtime`: 出版时间字符串

```

        owner_id: 拥有者 id
        ...
    }{...}]
searchUser;[{
    Id:用户 id
    name: 书房名称
    Images:书房照片
}]常理书房集合中只会有一个
}

```

## 43、即时通讯--获取好友列表

接口说明：点击我的好友查询，好友集合

请求方法：post、get

请求 URL: <http://localhost:8080/wangyi/friendsList>

参数：

userId;//用户 id

返回值说明：[{

```

    id: 好友 id
    image: 头像图片
    nickname: 用户昵称
    ...
}]

```

## 44、即时通讯--搜索好友

接口说明：输入用户 id 搜索用户

请求方法：post、get

请求 URL: <http://localhost:8080/wangyi/selectUserOne>

参数：

userId;//查找的用户 id

返回值说明：[{

```

    id: 好友 id
    image: 头像图片
    nickname: 用户昵称
    ...
}]

```

## 45、即时通讯--添加好友

接口说明：输入用户 id 搜索用户

请求方法：post、get

请求 URL: <http://localhost:8080/wangyi/selectUserOne>

参数：

accid 发起者的 id

faccid 接受者 id

type1 直接加好友, 2 请求加好友, 3 同意加好友, 4 拒绝加好友

msg 加好友对应的请求消息, 第三方组装, 最长 256 字符

返回值说明：{

    success:boolean 类型//是否成功 tru 表示操作成功

    Message:字符串//提示信息

}

## 46、即时通讯--删除好友

接口说明：输入用户 id 搜索用户

请求方法：post、get

请求 URL: <http://localhost:8080/wangyi/deleteFriend>

参数：

accid 发起者的 id

faccid 接受者 id

返回值说明：{

    success:boolean 类型//是否成功 tru 表示操作成功

    Message:字符串//提示信息

}

## 47、即时通讯--发送普通消息

接口说明：输入用户 id 搜索用户

请求方法：post、get

请求 URL: <http://localhost:8080/wangyi/sendMsg>

参数：

SendMsg sendMsg 对象{

    from 必须参数 发送者 accid, 用户帐号

    ope 必须参数 0: 点对点个人消息, 1: 群消息（高级群），其他返回 414

    to 必须参数 ope==0 是表示 accid 即用户 id, ope==1 表示 tid 即群 id

    type 必须参数 0 表示文本消息, 1 表示图片, 2 表示语音, 3 表示视

    频, 4 表示地理位置信息, 6 表示文件,

**body** 必须参数 请参考下方消息示例说明中对应消息的 **body** 字段，最大长度 5000 字符，为一个 JSON 串

```
}
```

返回值说明: {  
    **success**:boolean 类型//是否成功 **tru** 表示操作成功  
    **Message**:字符串//提示信息  
}

## 48、即时通讯--查询消息

接口说明: 输入用户 id 搜索用户

请求方法: post、get

请求 URL: <http://localhost:8080/wangyi/querySessionMsg>

参数:

**from** 必须参数 查询者 accid, 用户帐号

**to** 另一个用户 id

返回值说明: [{

image:头像

nickname:昵称消息

msgs:消息内容

```
}]
```

## 49、高德地图--根据坐标添加一条记录

接口说明: 添加一条位置信息到地图

请求方法: post、get

请求 URL: <http://localhost:8080/poi/creatDataOne>

参数:

Data data 对象{

\_name; //数据名称

\_location; //当 loctype=1 时, 必填规则: 经度, 纬度, 经纬度支持到小数点后 6 位 格式示例: 104.394729, 31.125698

coordtype; //可选值: 1: gps; 2: autonavi; 3: baidu 您可输入 coordtype=1, 或者 autonavi

\_address; //地址当 loctype=2 时, 必填

motto; //自定义列可选

userId; //用户 id

```
}
```

返回值说明: {  
    **success**:boolean 类型//是否成功 **tru** 表示操作成功



Message:字符串//提示信息

```
}
```

## 50、高德地图--根据位置添加一条记录

接口说明：添加一条位置信息到地图

请求方法：post、get

请求 URL: <http://localhost:8080/poi/creatDataOne2>

参数:

Data data 对象{

  \_name; //数据名称

  \_location; //当 loctype=1 时，必填规则：经度,纬度，经纬度支持到小数点后 6 位 格式示例：104.394729,31.125698

  coordtype; //可选值：1: gps;2: autonavi;3: baidu 您可输入 coordtype=1,或者 autonavi

  \_address; //地址当 loctype=2 时，必填

  motto; //自定义列可选

  userId; //用户 id

```
}
```

返回值说明：{

  success:boolean 类型//是否成功 tru 表示操作成功

  Message:字符串//提示信息

```
}
```

## 51、高德地图--修改位置信息

接口说明：根据位置 id 更新一条位置信息

请求方法：post、get

请求 URL: <http://localhost:8080/poi/creatDataOne2>

参数:

Data data 对象{

  id; 需要修改某一条来记录的 id

  \_name; //数据名称

  \_location; //当 loctype=1 时，必填规则：经度,纬度，经纬度支持到小数点后 6 位 格式示例：104.394729,31.125698

  coordtype; //可选值：1: gps;2: autonavi;3: baidu 您可输入 coordtype=1,或者 autonavi

  \_address; //地址当 loctype=2 时，必填

  motto; //自定义列可选

  userId; //用户 id

```
}
```

返回值说明: {  
    **success**:boolean 类型//是否成功 **tru** 表示操作成功  
    **Message**:字符串//提示信息  
}

## 52、高德地图--周边检索

接口说明: 根据坐标查询周边

请求方法: post、get

请求 URL: <http://localhost:8080/poi/surroundingSearch>

参数:

**center**: 经纬度用,号分割 113.663856,34.797789

返回值说明: {  
    **"count"**: "8", //符合的记录条数  
    **"info"**: "OK", //链接成功  
    **"infocode"**: "10000",  
    **"status"**: 1, //状态码  
    **"datas"**: [{  
        **"\_id"**: "16", //这个位置的 id  
        **"\_location"**: "113.660555,34.800156", //坐标  
        **"\_name"**: "你猜我猜不猜", //名字  
        **"\_address"**: "河南省郑州市金水区", //位置  
        **"userId"**: "1", //关联的用户 id  
        **"motto"**: "", //自定义的保留字段  
        **"\_createtime"**: "2018-08-01 14:37:57", //创建时间  
        **"\_updatetime"**: "2018-08-02 15:29:11", //最后修改时间  
        **"\_province"**: "河南省",  
        **"\_city"**: "郑州市",  
        **"\_district"**: "金水区",  
        **"\_distance"**: "401",  
        **"\_image"**: []  
    }, {...}]  
}

